

Amendments to the Claims:

This listing of claims replaces all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A system located on a client device for communicating data packets between the client device and a host system, the system comprising:

a protocol server module structured and arranged to:

terminate a communication session that uses a first protocol and that is intended to enable communications between a source and a destination, wherein the source is one of the client device operating system protocol stack and the host system and the destination is one of the client device operating system protocol stack and the host system but differs from the source,

translate data packets from the source between the first protocol and a second protocol that differs from the first protocol,

receive configuration data assigned by the host system, and

transport the data packets having the second protocol to the destination using the configuration data assigned by the host system; and

a controller module that is logically connected to the protocol server module and that is structured and arranged to control communications between the client device operating system protocol stack, the protocol server module, and the host system.

2. (Original) The system of claim 1 wherein the data packets include an encapsulation and the protocol server module translates the data packets by removing the encapsulation from the data packets.

3. (Currently Amended) The system of claim 1 wherein the protocol server module translates the data packets by encapsulating the data packets using any one of several communication protocols that differs from the ~~original~~ first protocol.

4. (Currently Amended) The system of claim 2 wherein the protocol server module translates the data packets to further include encapsulating the data packets using any one of several communication protocols that differs from the ~~original~~ first protocol.

5. (Original) The system of claim 1 wherein the client device operating system protocol stack supports PPP.

6. (Original) The system of claim 5 wherein the protocol server module includes a PPP server module located on the client device.

7. (Original) The system of claim 6 wherein the PPP server module is structured and arranged to terminate a PPP communication session between the client device operating system protocol stack and the host system.

8. (Original) The system of claim 6 wherein the PPP server module is structured and arranged to negotiate a PPP communication session with the client device operating system protocol stack.

9. (Original) The system of claim 1 wherein the protocol server module and the controller module perform transparent to a sender of the data packets.

10. (Original) The system of claim 1 wherein the protocol server module is structured and arranged to enable collection of data for error checking.

11. (Original) The system of claim 1 wherein the protocol server module is structured and arranged to filter the data packets prior to transporting the data packets to the destination.

12. (Original) The system of claim 1 further comprising a virtual modem adapter logically connected between the client device operating system protocol stack and the protocol server module.

13. (Original) The system of claim 1 wherein the data packets include layer three data packets.

14. (Currently Amended) A method for communicating data packets between a client device and a host system through a network, the method comprising:

at the client device, terminating a communication session that uses a first protocol and that is intended to enable communications between a source and a destination, wherein the source is one of a client device operating system protocol stack and the host system and the destination is one of the client device operating system protocol stack and the host system but differs from the source;

translating data packets from the source between the first protocol and a second protocol that differs from the first protocol;

receiving configuration data assigned by the host system; and

transporting the data packets having the second protocol to the destination using the configuration data assigned by the host system.

15. (Original) The method as in claim 14 wherein the data packets include an encapsulation and translating the data packets includes removing the encapsulation from the data packets.

16. (Currently Amended) The method as in claim 14 wherein translating the data packets includes encapsulating the data packets using any one of several communication protocols that differs from the ~~original~~ first protocol.

17. (Currently Amended) The method as in claim 15 wherein translating the data packets further includes encapsulating the data packets using any one of several communication protocols that differs from the ~~original~~ first protocol.

18. (Original) The method as in claim 14 wherein:  
terminating the communication session includes using a protocol server module to terminate the communication session;

translating the data packets includes using the protocol server module to translate the data packets from the source between the first protocol and the second protocol; and

transporting the data packets includes using the protocol server module to transport the data packets having the second protocol to the destination.

19. (Original) The method as in claim 18 wherein the client device operating system protocol stack supports PPP.

20. (Original) The method as in claim 19 wherein the protocol server module includes a PPP server module located on the client device.

21. (Original) The method as in claim 20 wherein terminating the communication session includes using the PPP server module to terminate a PPP communication session between the client device operating system protocol stack and the host system.

22. (Original) The method as in claim 20 wherein terminating the communication session includes using the PPP server module to negotiate a PPP communication session with the client device operating system protocol stack.

23. (Original) The method as in claim 18 further comprising using a virtual modem adapter to interface between the client device operating system protocol stack and the protocol server module.

24. (Original) The method as in claim 14 wherein terminating the communication session, translating the data packets, and transporting the data packets occur transparently to a sender of the data packets.

25. (Original) The method as in claim 14 further comprising enabling collection of data for error checking.

26. (Original) The method as in claim 14 further comprising filtering the data packets prior to transporting the data packets to the destination.

27. (Original) The method as in claim 14 wherein the data packets include layer three data packets.

28. (Currently Amended) A system located on a client device for communicating data packets between the client device and a host system, the system comprising:

a protocol server module structured and arranged to:

terminate a communication session between a source and a destination,  
wherein the source is one of a client device operating system protocol stack and the host system and the destination is one of the client device operating system protocol stack and the host system but differs from the source,

receive configuration data assigned by the host system, and

transport data packets to the destination through a network using any one of several communication protocols using the configuration data assigned by the host system; and

a controller module that is logically connected to the protocol server module and that is structured and arranged to control communications between the client device operating system protocol stack, the protocol server module, and the host system.

29. (Original) The system of claim 28 wherein the protocol server module is structured and arranged to translate the data packets prior to transporting the data packets.

30. (Original) The system of claim 29 wherein the data packets include an encapsulation and the protocol server module translates the data packets by removing the encapsulation from the data packets.

31. (Currently Amended) The system of claim 29 wherein the protocol server module translates the data packets by encapsulating the data packets using any one of several communication protocols that differs from ~~[[the]]~~ an original protocol.

32. (Currently Amended) The system of claim 30 wherein the protocol server module translates the data packets to further include encapsulating the data packets using any one of several communication protocols that differs from ~~[[the]]~~ an original protocol.

33. (Original) The system of claim 28 wherein the client device operating system protocol stack supports PPP.

34. (Original) The system of claim 33 wherein the protocol server module includes a PPP server module located on the client device.

35. (Original) The system of claim 34 wherein the PPP server module is structured and arranged to terminate a PPP communication session between the client device operating system protocol stack and the host system.

36. (Original) The system of claim 34 wherein the PPP server module is structured and arranged to negotiate a PPP communication session with the client device operating system protocol stack.

37. (Original) The system of claim 28 wherein the protocol server module and the controller module perform transparent to a sender of the data packets.

38. (Original) The system of claim 28 wherein the protocol server module is structured and arranged to enable collection of data for error checking.

39. (Original) The system of claim 28 wherein the protocol server module is structured and arranged to filter the data packets prior to transporting the data packets to the destination.

40. (Original) The system of claim 28 further comprising a virtual modem adapter logically connected between the client device operating system protocol stack and the protocol server module.

41. (Original) The system of claim 28 wherein the data packets include layer three data packets.

42. (Currently Amended) A method for communicating data packets between a client device and a host system through a network, the method comprising:

at the client device, terminating a communication session between a source and a destination, wherein the source is one of a client device operating system protocol stack and the host system and the destination is one of the client device operating system protocol stack and the host system but differs from the source;

receiving configuration data assigned by the host system; and

transporting data packets to the destination through the network using any one of several communication protocols using the configuration data assigned by the host system.

43. (Original) The method as in claim 42 further comprising translating the data packets prior to transporting the data packets.

44. (Original) The method as in claim 43 wherein the data packets include an encapsulation and translating the data packets includes removing the encapsulation from the data packets.

45. (Currently Amended) The method as in claim 43 wherein translating the data packets includes encapsulating the data packets using any one of several communication protocols that differs from ~~[[the]]~~ an original protocol.

46. (Currently Amended) The method as in claim 44 wherein translating the data packets further includes encapsulating the data packets using any one of several communication protocols that differs from ~~[[the]]~~ an original protocol.

47. (Original) The method as in claim 42 wherein:  
terminating the communication session includes using a protocol server module to terminate the communication session; and

transporting the data packets includes using the protocol server module to transport the data packets to the destination through the network using any one of several communication protocols.

48. (Original) The method as in claim 47 wherein the client device operating system protocol stack supports PPP.

49. (Original) The method as in claim 48 wherein the protocol server module includes a PPP server module located on the client device.

50. (Original) The method as in claim 49 wherein terminating the communication session includes using the PPP server module to terminate a PPP communication session between the client device operating system protocol stack and the host system.

51. (Original) The method as in claim 49 wherein terminating the communication session includes using the PPP server module to negotiate a PPP communication session with the client device operating system protocol stack.

52. (Original) The method as in claim 47 further comprising using a virtual modem adapter to interface between the client device operating system protocol stack and the protocol server module.

53. (Original) The method as in claim 42 wherein terminating the communication session and transporting the data packets occur transparently to a sender of the data packets.

54. (Original) The method as in claim 42 further comprising enabling collection of data for error checking.

55. (Original) The method as in claim 42 further comprising filtering the data packets prior to transporting the data packets to the destination.



56. (Original) The method as in claim 42 wherein the data packets include layer three data packets.

57. (New) The system of claim 1 wherein the configuration data includes IP configuration data.

58. (New) The system of claim 1 wherein the configuration data is received without manual user intervention.

59. (New) The method as in claim 14 wherein the configuration data includes IP configuration data.

60. (New) The method as in claim 14 wherein the configuration data receiving the configuration data includes receiving the configuration data without manual user intervention.

61. (New) The system of claim 28 wherein the configuration data includes IP configuration data.

62. (New) The system of claim 28 wherein the configuration data is received without manual user intervention.

62. (New) The method as in claim 42 wherein the configuration data includes IP configuration data.

64. (New) The method as in claim 42 wherein receiving the configuration data includes receiving the configuration data without manual user intervention.